A Review of Film Editing Techniques for Digital Games

Rémi Ronfard

▶ **To cite this version:**

Rémi Ronfard. A Review of Film Editing Techniques for Digital Games. Workshop on Intelligent Cinematography and Editing, May 2012, Raleigh, United States. hal-00694444

HAL Id: hal-00694444
https://inria.hal.science/hal-00694444

Submitted on 4 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Review of Film Editing Techniques for Digital Games

Remi Ronfard
INRIA, LJK, Université de Grenoble, France
remi.ronfard@inria.fr

1. INTRODUCTION

Automated film editing involves the generation of the position, orientation, motion and selection of virtual cameras in interactive 3D graphics applications. There is a pressing demand for techniques to assist and automate the control of virtual cameras in the computer games industry where the rapid development of personal computers and high performance consoles has led to substantial improvements in the visual fidelity of games. The goal of this survey is to characterize the spectrum of applications that require automated film editing, present a summary of state-of-the-art models and techniques, and identify both promising avenues and hot topics for future research

2. CINEMATOGRAPHY AND EDITING

One fundamental part of cinematography, as outlined in Maschielli's $5C$'s of cinematography [14] is to provide shots that can easily be edited together. In the early days of cinema, the interplay between cinematography and editing was a matter of trial and error. As noted by Barry Salt [20], it took several years before cinematographers and editors understood the "exit left enter right" editing rule. Before that, the rule was usually obeyed because it appeared to work better in most cases. But the "wrong" solution was still used from time to time. When it finally became clear what the "right" solution was, cinematographers stopped shooting the alternate solution because they knew it was useless. After more than a century of cinema, good professional cinematographers have thus "internalized" the rules of editing in such a way that they can avoid shots that will not cut together.

In games, we are probably still at an earlier stage because it is not yet quite clear how the rules of cinematography should translate for an interactive game, which is a very different situation from a movie.

In computer graphics, the camera is controlled by animators. A good professional animator should have a similar sense of which shots will cut together. When this is not the case, the editor is left with fewer or no options. As a result, the scene may have to be shot again from another angle. This is usually not a problem because it is easy (and cheap) to do so. When implementing automated systems, it is important to take the rules of editing into account in the early stages of planning and controlling the camera. Otherwise, a lot of effort will be wasted on attempting to edit shots that "do not cut together". This will be examined in depth in Section 3.

In traditional cinematography, cutting can be taken into account by following one of several working practises. We mention three of them.

1. **Cutting in the head** means that the director has already decided very precisely every single shot, usually in the form of a storyboard. In that case, it suffices to shoot each action or *beat* in the screenplay from a single viewpoint. Textbooks in film-making warn against the dangers of the method because it cannot recover easily from errors in planning.

   This approach is very suitable for real-time applications. It consists in planning the editing first, resulting in a list of shots that can then be rendered *exactly as planned* following the timeline of the final movie.

   One drawback of that approach is that the animation itself cannot always be predicted in all its actual details. As a result, it may be difficult to plan exactly *when to cut* from shot to shot.

2. **Three-take technique** A common variant of "cutting in the head" consists in shooting a little more of the action from each planned camera position. As a result, each action is shot from three camera positions - one according to the shot list, one from the immediately previous viewpoint and one from the next viewpoint.

   This has the advantage that the exact cutting point can be resolved at a later stage.

3. **Master-shot technique** Another common practice consists in planning all the camera works for shooting the scene in one continuous take - the "master shot" - and then adding shots of various sizes to show the details of the action in various sizes (close-ups and medium shots). Editing can then more carefully prepared by insuring that all those shots will cut nicely with the master shot, resulting in a typical sequence of "Master-Closeup-Master-Closeup", etc.

Note that those techniques are very useful in practice because they are more general than "film idioms" where the camera positions are prescribed once and for all.

3. AUTOMATIC CAMERA EDITING

This section covers the approaches that draw on a theory of film editing for planning and performing camera placement and composition. Here scenes are described in terms of actions and communicative goals that must be translated into successive shots. Cutting between cameras adds considerable freedom in the focalization and order of presentation of the visual material. Cutting between cameras also introduces constraints. We review the most important constraints and corresponding rules (180 degree rule, 60 degree rule) and explain how they can be expressed and solved algorithmically. Then, we review the principles that can be used to evaluate the quality of a shot sequences and the algorithmic strategies that can be used to solve for the best sequence. Finally, we review the strengths and limitations for some of the existing systems proposed for real-time, live-editing [He96,Funge98] as well as offline, post-production editing [Elson07] and sketch promising future directions for research in this area.

Automatic film editing has a long history, dating back at least to Gilles Bloch's PhD thesis in 1986 [1]. In this section, we present both procedural and declarative approaches. A procedural approach to movie editing builds an explicit solution. A good example of that is the Virtual Cinematographer system (VC) where each idiom is implemented as finite state machine. A reactive approach is essentially a procedural approach where multiple courses of events can be taken into account. A declarative approach states the constraints and rules and lets a separate solver find a solution that meets all the constraints, and/or maximizes a measure of quality.

3.1 Editing rules and constraints

It is important to understand the motivation between the so-called "rules of editing". Most of them are in fact constraints. What that means is that it may not be possible to cut from any two arbitrary cameras. Why not ? Because some transitions may provoke *false inferences*. For a cut between two shots to work, it is fundamental that it does not break the logic of human perception.

Psychologists dÂŠYdewalle and Vanderbeeken offer a useful classification of editing errors [8]. Editing errors of the "first order" are small displacements of the camera or image size, disturbing the perception of apparent movement and leading to the impression of jumping. Editing errors of the "second order" are violations of the spatial-cognitive representation of the 3-D scene. One example is the 180-rule violation, where the camera crosses the line between two actors and as a result the actors appear to swap positions. Another example is the motion continuity violation, when the camera crosses the line of an actor's movement and as a result the actor appears to change directions. Editing errors of the "third-order" are when successive shots have too little in common to be integrated into a single chronological sequence of events.

An important part of automated movie editing consists in preventing editing errors of all orders. But that is of course not the entire story because there are still infinitely many "correct" camera pairs that can be cut together at any given time. A second part of automated editing is therefore to evaluate *when* to cut to *which* shot.

The classical Hollywood concept of editing [14] recommends that successive shots should minimize perceptually disruptive transitions. The modern viewpoint [9] stresses the consistency of the narrative structure which overrule disturbing transitions, as attention will primarily be directed to grasping the succession of significant events in the story. A good computational theory of film editing should probably stand in the middleground between those two viewpoints. On the one hand, it is difficult to get a good model of "perceptually disruptive transitions". At best, a computational model may be expected to avoid the most obvious mistakes, still leaving a large number of possibilities. On the other hand, the narrative structure of an animated scene may not always be easily uncovered, again leaving multiple choices.

Few editors have written about their art with more depth than Walter Murch [16]. In his book, he introduces a Rule of Six with six layers of increasing complexity and importance in the choice of how and when to cut between shots:

Three-dimensional space of action. Respect of 3-D continuity in the real world: where people are in the room and their relations to each other (accounts for only 4 % of what makes a good cut)

Two-dimensional space of screen. Respect of 2D continuity. Where people appear on the screen. Where the lines of action, look, movement project on the screen. (5 %)

Eye-trace. Respect of the audience's focus of interest before and after the cut. (7 %)

Rhythm. Cut at a moment which is both right and interesting. (10 %)

Story. Cut in a way that advances the story. (23 %)

Emotion. Cut in a way that is true to the emotion of the moment. (accounts for 51 % of what makes a good cut).

In 3-D animation, the three-dimensional space of action is always in continuity as long as we perform live editing. So we only really need to be concerned with the other five criteria. We can attempt to build a computational theory of film editing based on this reduced rule of five if we know how to evaluate each of the five criteria AND find a consistent way to rank possible cuts and shots using a combination of them.

3.1.1 Two-dimensional continuity.

Two-dimensional continuity is easiest to evaluate by computer. All the programmer has to do is project the various lines (of action, of looks, of movements, etc) to the camera plane and check that they remain consistent. This is a direct application of projective geometry.

Two-dimensional continuity can be insured by adhering to the following rules of the so-called *classical continuity style*:

Line of action The relative ordering of characters must remain the same in the two shots.

This is the basis for the 180 degree rule, which forbids cuts between cameras situated across a line between the two characters - the line of action.

Screen continuity Characters who appear in both shots must not appear to jump around too much.

Motion continuity Moving characters who appear in both shots must appear to move in the same screen direction.

This is the basis for another variant of the 180 degree rule, which forbids cuts between cameras situated across a line along the actor's trajectory - the line of action in that case.

Motion continuity also requires that the screen position of the actor in the second shot should be "ahead", rather than "behind"

Jump cut Characters who appear in both shots must not appear to jump around too little.

Small changes in screen coordinates are interpreted as actor movements, rather than camera changes, as an effect of human perception. They should be avoided, or used systematically to obtain a stylistic effect (Godard).

Look The gaze directions of characters seen in separation should match. If they are looking at each other, their images should also be looking at each other. If the two characters are NOT looking at each other, their images should NOT be looking at each other

Distance The sum of apparent distances to two characters shown in separation should be at least twice the actual distance between them (as if the two images were taken from the same camera position). This prevents the use of close-ups for two characters very far apart.

Size The shot size relative to a character should change smoothly, rather that abruptly.

Cutting from a long shot directly to a close-up makes it harder for the viewer to understand the relation between the two shots. Instead, the editor should prefer to first cut to a medium-shot, then to a close-shot.

It is important to realize that the rules can be checked by direct computations in screen space (through-the-lens). They typically do not require knowledge of world coordinates.

3.1.2 Eye-trace.

Eye-trace refers to the expected trajectories of the eyes of the audience. Where on the screen is the audience looking in the first shot ? What happens there during the cut ? Where will the audience look in the second shot ?

A popular heuristic is to use the actors' eyes in the image. This is a well established principle confirmed by many film editors. But predicting where the audience is looking remains hard even for editors. In the context of stereoscopic 3-D Film director James Cameron (who also edits his own movies) phrased it as follows: "You can only converge to one image plane at a time – make sure it is the place the audience (or the majority of the audience) is looking. If it's Tom Cruise smiling, you know with 99% certainty where they're looking. If it's a wide shot with a lot of characters on different depth-planes doing interesting things, your prediction rate goes down."

Current research in vision science attempts to predict the focus of attention in an image, based on the computation of local image features. The most established theory is the "saliency-based" model of Itti and Koch at Caltech [11]. Their model was used by Santella et al. for the purpose of evaluating the composition while cropping and reframing images [22]. Their conclusion was that better predictions were obtained by considering the eyes and gaze of people in the image.

3.1.3 Rhythm.

Rhythm refers to the tempo of the scene (how fast the film is cut). But we should be aware that the perceived duration of a shot depends on its content. Thus a shot that we have already seen many times will seem to last longer than it really is. A close-up will also seem to last longer than it really is. We should cut from any given shot only after the audience has been able to fully see what we intend them to see. We should also cut before the shot becomes redundant or boring.

One further complication is that the perceived length of a shot depends on its size, its novelty and the intensity of the action. Thus, a close-up will be perceived as taking longer than a long shot. A recurring shot will be perceived as taking longer than a new shot. And a shot of a static scene will be perceived as taking (much) longer than a shot of a fast action. A reasonable approximation may be to set the average shot length as a function of shot size, so that close-ups are cut faster and long shots are cut slower. This is a reasonable first approximation.

Another important factor is to choose a *natural* distribution of shot durations. Automated editing should not "get in the way". As a very simple illustrative example, cutting at regular intervals (as with a metronome) can be very annoying because it distracts the viewer from the experience of the movie. Cutting shots with randomized durations is usually a better idea. Even better editing can be computed by following the distribution of shot durations in real movies.

Film scholars Barry Salt [20] and David Bordwell [3] (among others) have extensively studied shot durations in cinema and found it to be an important parameter of film style. An empirical finding by Barry Salt is that the distribution of shot durations in a movie sequence is correctly represented by a log-normal distribution. This is also the distribution of sentence lengths in a book chapter. This is non-symmetric distribution with a smaller probability for very short durations and a relatively larger probability for longer shot durations.

What is important is to set the editing rhythm by choosing an *average shot length* or ASL for the sequence, and cut according to a log-normal distribution. We can fine-tune the rhythm by also choosing the variance σ^2 of the shot lengths.

3.1.4 Story advancement.

Story advancement can be measured by checking that all changes in the story line are correctly presented in the image. Thus, actors should only change places on-screen (not off-screen). We should see (or hear) their reactions. We should see entrances and exits of all characters. We should see them when they sit down or stand up, when they dress or undress, when then they put on or take off their hats, etc. Of course, real directors and editors break this rule all the times, with interesting effects. But it seems to be a safe bet to adopt the rule that the best editing is the one that *presents the entire action in the scene from the best angle at all times.*

An even stronger principle was proposed by Hitchcock in an interview with Truffaut [23]. "Screen size and visibility of actors and objects should be proportional to their importance in the plot at any given time (Hitchcock principle). This is useful principle to keep in mind because it allows the programmer to define mathematically what makes a good editing. Computing the screen size and visibility of actors and objects in a shot is the easy part. Computing their importance in the plot is the really difficult part.

In a scripted sequence, it seems reasonable to assume that the scripted actions are all equally important. Thus at any given time, the importance of actors and objects can be approximated as the number of actions in which they are taking part, divided by the total number of actions being executed in the scene at that time. Other approximations are of course possible. For instance, it may be preferable to assign all the attention to a single action at all times. This may be implemented with a "winner takes all" strategy.

3.1.5 Emotion.

Emotion is hardest to evaluate. There is a large body of research being done in neuroscience on emotion. They distinguish between primitive emotions, such as surprise, fear, laughter, etc. whose action is very fast; primitive moods, such as sadness or joy, whose action is much slower; and learned, cognitive affects such as love, guilt, shame, etc.

For the purpose of editing, evaluating the emotional impact of any given shot or cut appears to be very difficult. Emotional cues can be received from the screenplay or from the director's notes. They assert which emotions should be conveyed at any given point in time. Given such emotional cues, we can then apply simple recipes such as separating actors or showing them closer together; changing editing rhythm to show increasing or decreasing tension; changing shot sizes to show increasing or decreasing tension; using lower camera angles to show ceilings and feel oppression; using higher camera angles to hide ceilings and feel freedom; using longer lenses to slow down actor movements and isolate them from the background; using wider lenses to accelerate actor movements and put them in perspective, etc. How simple should those strategies be ? Too simple a solution may look foolish. Too complicated solution may be out of reach.

The emotional content of a shot or a transition between shots has been little explored [21, 28] and is a promising avenue for future research in cinematography.

After having explained the theory of editing, we now turn to actual implementations of working systems. We review procedural and declarative approaches separately.

3.2 Procedural approaches

The Virtual Cinematographer by He et al. [10] relies on the use of film idioms, which are recipes for obtaining good framing and editing in a given situation. The general approach is similar to the old-fashioned AI principle of case-based reasoning - if a conversation starts in a game, use the conversation idiom; if a fight start, use the fight idiom; etc.

Each idiom has two components - a set-up (blocking) of the cameras relative to the actors; and a state machine for switching automatically between cameras in that setup. This is a powerful paradigm, that easily allows for gradually building up a complex cinematography system from simple building blocks.

Each idiom is very easy to program - the set-up of the cameras is defined in terms of world coordinates - relative to the actors. The VC takes as input strings of simple sentences : SUBJECT+VERB+OBJECT representing the action taking place in the scene. The VC also takes as input a continuous stream of bounding boxes and orientation, representing the relative geometric positions and orientations of the virtual actors, objects and scene elements.

Idioms are usually chosen based on the next action string. More complex editing patterns can also be achieved by defining hierarchical state machines, encoding the transitions between idioms.

While powerful, this scheme has yet to demonstrate that it can be used in practical situations. One reason may be that there is a heavy burden on the application programmer, who must encode all idioms for all narrative situations. Another reason may be that the resulting editing may be too predictable.

In a finite state machine, the switching of a camera is triggered by the next action string. This may have the undesirable effect that the switching becomes too predictable. A good example is the "dragnet" style of editing [16] where the camera consistently switches to a close-up of the speaker on each speaker change; then back to a reaction shot of the other actors being spoken to. This can become especially annoying when the speakers alternate very quickly.

While it is possible to use the dragnet style of editing as a separate film idiom, this causes the number of idiom to explode since every configuration can be filmed in dragnet style. A better solution separates the camera set-ups from the state machines - for each set-up, different styles can then be encoded with different state machines. But the same "style" must still be separately re-encoded for each set-up.

It is not obvious how to "generalize" film idioms. This is an open problem for procedural approaches.

3.3 Declarative approaches

In the beginning, automatic editing was attempted with traditional, rule-based systems.

IDIC by Sack and Davis [19] was one of the first systems to attempt automatic film editing from annotated movie shots. Mostly a sketch of what is possible, it was based on the general problem solver (GPS), a very simple forward planner [18].

"Declarative Camera Control for Automatic Cinematography" is a much more elaborate attempt at formalizing the editing of an animated movie, this time using modern planning techniques [4]. In that paper, idioms are not described in terms of cameras in world coordinates but in terms of shots in screen coordinates, through the use of the DCCL language. DCCL is compiled into a film tree, which contains all the possible editings of the input actions. Actions are represented as subject-verb-object triples. As in the Virtual Cinematographer companion paper, the programming effort for implementing an idiom is important.

Jhala and Young have used text generation techniques to automatically edit shots together using "plan operators" [12]. In another paper, Jhala and Young have used examples from the movie "The Rope" by Alfred Hitchcock to emphasize stronger requirements on how the story line AND the director's goal should be represented to an automatic editing system [13]. They use Crossbow, a partial order causal link planner, to solve for the best editing, according to a variety of strategies, including maintaining tempo and depicting emotion. They do not attempt to combine those strategies and instead prefer to demonstrate the capability of their solver to present the same sequence in different editing styles.

Miyazaki et al. describe a complete film-making production system [25, 26]. They model the scene graph in CLIPS/COOL and define rules for choosing cameras and editing them. But they are restricted to common idioms.

Kennedy and Mercer use the LOOM knowledge representation language to encode different communicative acts in the rhetorical structure theory. By mapping the story-line into communicative goals, stated in terms of themes and moods, they are able to plan the choice of camera and editing. They discuss "inductive" and "deductive" approaches of characters, using Zettl as a reference [30].

Friedman and Feldman present another knowledge-rich approach for editing sitcoms [7].

AI-based approaches present an excellent overview of many important aspects of automated film editing, but the results are not always convincing for lack of a sufficient integration with advanced camera control techniques. Another drawback of the AI-based approach is that it requires an in-depth semantic analysis of the storyline, which is not always readily available in practical applications, especially in real-time games. More importantly, those methods usually return a (usually large) list of possible solutions, even in simple cases. As a result, they usually do not scale very well with larger vocabularies of plot actions, films idioms and shot categories.

3.4 Optimization approaches

To overcome the problems of procedural and AI-based declarative approaches, it seems natural to rephrase the editing problem as an optimization problem. In this section, we revisit the editing constraints listed above and illustrate how they can be used to build a quality function.

Let us review the common case of a dialog scene between two actors. We are given a sequence of time intervals, each of which may include actions performed by the two characters A and B.

$$(a_1, b_1, t_1), (a_2, b_2, t_2), ..., (a_n, b_n, t_n)$$

A solution to the automatic editing problem is a sequence of shots

$$(c_1, s_1, t_1), (c_2, s_2, t_2), ..., (c_n, s_n, t_n)$$

where each shot s_i is taken by camera c_i starting at time t_i. Cuts occur whenever the camera changes between successives intervals. Reframing actions occur when the camera remains the same but the shot descriptions change. Transitions between the same shots result in longer shots and we can write the duration of the shot Δ_i.

Most optimization approaches compute the cost of a sequence from three types of preferences. They can use stylistic preferences on shots; preference on which shot to use for each action; and preferences on how to cut from one shot to the next one. This can be include stylistic preferences as well. For example, we can prefer shots whose durations are modeled by a log-normal distribution with average shot length (ASL) m and standard deviation σ. The cost associated with a shot of length Δt can then be expressed as

$$C(\Delta) = -\log p(\Delta) = \log \Delta + \frac{(\log \Delta - \log m)^2}{2\sigma^2}$$

Other style parameters are the desired ratios of long shots, medium shots and close shots; the relative importance of the two characters, measured by their screen-time; and the relative importance between verbal and non-verbal action.

As stated previously, an important goal in camera control and editing is to show actions of the characters appropriately. In each segment, the editor reads the actions a_i and b_i performed by the two actors and the corresponding tables for shot preferences. Those tables can easily be pre-computed for general action categoriessuch as speech actions, facial expressions, hand gestures and pointing gestures. THis gives a ranking of shot choices for each category.

The editor must also able to express preferences for the transitions between a shot s_i and a shot s_j based on film grammar. Examples of preferences are - keep each character on the same side of the screen across a cut. Avoid cutting between two-shots. Avoid cutting from a long-shot to a close-shot. Prefer cutting between shots of the same size (long, medium of close). Etc. A general method for building such as table is by *counting editing errors of all orders*. Again, this can be captured in an $n \times n$ table.

The problem is then of finding the sequence with the best

ranking. With a Markov assumption, finding a sequence of shot transitions that maximizes the quality function can be very efficiently implemented by dynamic programming. That approach is taken by Elson and Riedl [6]. Note that this precludes the use of shot durations. A semi-Markov or "segment" model is needed for enforcing shot durations. That approach was introduced in Xtranormal's "magicam" system [17]. Higher-order Markov models would be useful to implement repetition, which is a very powerful cinematic technique [24], not easily taken into account with a Markov model.

A promising approach that combines AI-based planning with optimization is hierarchical task network (HTN) planning with preferences [29], which has been used in game AI to solve military team planning. While we are not aware of it being used for planning cinematography, it appears to be a likely candidate for future work in this area.

3.5 Applications

3.5.1 Interactive storytelling

Interactive storytelling promises to become a hybrid between film and game, with a very strong need for fully automated real-time cinematography and editing, so that all possible navigation paths through the "story graph" generate movies that are aesthetically pleasing. FACADE by Mateas and Stern is a good example, although with a very simple cinematic look [15].

3.5.2 Automated movie production

Some systems go even beyond camera control and editing , towards fully automated movie production. In 1966, Alfred Hitchcock dreamed of ÂŞa machine in which heÂŠd insert the screenplay at one end and the film would emerge at the other end, complete and in colorÂŤ (Truffaut/Hitchcock, p. 330). In limited ways, this dream can be achieved by combining the staging of virtual actors with the techniques of camera control and editing described in this course. An example is the text-to-scene system by Xtranormal. This includes various declarative shots - one-shots and two-shots with a variety of camera angles and compositions. Camera placement is automated for declarative shots. Editing is fully automated and makes use of both declarative shots (idioms) and free cameras. This overcomes the traditional limitations associated with a purely idiom-based system. Visibility is taken into account through the use of "stages", i.e. empty spaces with unlimited visibility, similar to Elson and Riedl. Both systems use a simple algebra of "stages", i.e. intersections and unions of stages, allowing for very fast visibility computation against the static elements of the scene. Occlusion between actors is handled separately by taking pictures through the eyes of the actors. The text-to-scene system by Xtranormal is currently limited to short dialogue scenes, although with a rich vocabulary of gestures, facial expressions and movements. But we can expect future improvements and extensions to other scene categories, including action and mood scenes.

3.5.3 Machinima

Most machinima systems include some sort of camera control. For instance, MovieStorm by Short Fuze includes "through-the-lens" camera control with two types of cameras. A "free" camera can pan, tilt or roll around the camera axis in all directions. A "sticky" camera is attached to an actor's eye line. The camera can pan, tilt or roll "around the actor's eye axis" which is much more powerful. In principle, it is possible to implement sticky cameras on other targets, and multiple targets. This can be extended to compute dolly paths as well. For two-shots, the camera can move along a circle while keeping the two actors in any given screen position. Few machinima systems include support for editing. One may expect that this will change in the near future. This requires the ability to record an entire 4-D scene (3-D + time). Such replay functions are not usually offered in games. Dedicated applications such as Xtranormal State have it. Support for editing also requires "higher level" action descriptions of some sort. For machinima in a game engine, such high-level descriptions can in principle be inferred from the player's or puppeteer's actions. But player's intentions cannot easily be inferred from their movements. Non-player characters (NPC) have a more formalized vocabulary of intentions and actions. This can be used to motivate the cinematography. Cinematography is intimately related to game AI in that respect. The camera and editor "agents" must infer the players intentions and actions in order to correctly react to them. The main difference with other NPCs is that they pursue different goals. In essence, a camera is an NPC whose goal is to "follow and watch" other actors. In dedicated applications such as Xtranormal, The Sims or MovieStorm, the actors's movements are labeled with higher-level commands, including "looking", "speaking", "pointing", "sitting" or 'standing", etc. This is sufficient in principle to motivate the cinematography and editing. In addition, Movie Storm outputs a "movie script" inferred from the choice of actions. On the other hand, text-to-scene systems such as Xtranormal instead use the movie script as an input, and infer the sequence of actions to be performed by the virtual actors from the script.

4. DISCUSSION AND OPEN ISSUES

This final section discusses the problems related to the actual deployment of these techniques and directions for future research, including augmenting the expressiveness of camera control and switching techniques by considering cognitively well-founded perceptual and aesthetic properties of the shots, including framing and lighting ; extending camera models to include the control of other important cinematographic properties such as focus, depth-of-field (DOF) and stereoscopic 3-D depth (interaxial distance and convergence); and learning more general and varied camera control and editing idioms directly from real movies using a variety of data mining and machine learning techniques.

4.1 Perception and aesthetics

The state of the art in automatic framing (composition) and editing relies on a symbolic description of the view seen by the virtual camera. This is powerful, but important aspects of the image are not well taken into account. A possible avenue for future research lies in the possibility to perform *image analysis* directly from the virtual camera, to recover other important perceptual and/or aesthetics attributes of the image. This is especially important for lighting [5]. Other image attributes, such as the contrast between figure and background may be equally important [2].

s

4.2 Level of details

One as yet unexplored area for future research is the relation between cinematography and level-of-details modeling. During the early phases of pre-production and previz, a rough version of the scene with little details may be sufficient to do the blocking of the actors and the cameras, and even to generate an early version of the editing (a rough cut). The choices made in this stage result in a list of a few shots which need to be rendered at full resolution. Thus, only those parts of the scene that appear in the shot list really need to be modeled and rendered in full details. In practice, it is not easy to implement this because the animation must still appear realistic. Levels-of-details are still a problem for physics-based animation and AI.

4.3 Cinematic knowledge

Much of the current research remains limited to simple toy problems such as two-actor dialogues and fights. At this point, there has never been a convincing demonstration of a touching machine-generated love scene. Or a funny machine-generated comic scene. Or a frightening machine-generated horror scene.

This is the main challenge for this field. In the future, we expect that strategies will be taken to better reproduce the specific cinematography and editing styles needed for soccer and car-racing replays; machinima remakes of film noir, drama, soap and sitcom. Cinematography and editing styles can be hand-crafted through the use of idioms in a procedural system; or by choosing preferences in an optimization approach. One promising avenue of research for imitating movie styles is through the use of video data-mining, i.e. searching large annotated databases of movie examples [27].

4.4 Evaluation

Evaluation of automatic camera control and editing has been attempted by only a few researchers. The result seems to be that it is relatively easy to emulate an "amateur" cameraman or film editor, but very hard to emulate even a "modest" professional. In other words, empirical evaluations show that a professionally cinematographer and edited scene is always preferred to a machine-generated scene. But a machine-generated scene can be preferred (or found comparable) to an amateur-generated scene. Another possible evaluation criteria is *ease of use*. For example, it would be useful to compare the time needed for generating a satisfactory movie scene with different camera control and editing systems.

5. REFERENCES

[1] G. Bloch. *Eléments d'une machine de montage pour l'audio-visuel*. PhD thesis, T'elécom Paris, 1986.

[2] B. Block. *The Visual Story: Seeing the Structure of Film, Tv, and New Media*. Focal Press, 2001.

[3] D. Bordwell. *The Way Hollywood Tells It: Story and Style in Modern Movies*. University of California Press, 2006.

[4] D. B. Christianson, S. E. Anderson, L.-W. He, D. S. Weld, M. F. Cohen, and D. H. Salesin. Declarative camera control for automatic cinematography. In *Proceedings of AAAI '96 (Portland, OR)*, pages 148–155, 1996.

[5] M. S. El-Nasr. A user-centric adaptive story architecture: borrowing from acting theories. In *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 109–116, 2004.

[6] D. K. Elson and M. O. Riedl. A lightweight intelligent virtual cinematography system for machinima generation. In *AI and Interactive Digital Entertainment*, 2007.

[7] D. Friedman and Y. A. Feldman. Automated cinematic reasoning about camera behavior. *Expert Systems with Applications*, 30(4):694–704, May 2006.

[8] F. Germeys and G. dï£¡Ydewalle. The psychology of film: perceiving beyond the cut. *Psychological Research*, 71(4):458–466, 2007.

[9] J.-L. Godard. Montage, mon beau souci. *Les cahiers du cinéma*, 11(65), décembre 1956.

[10] L. He, M. Cohen, and D. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *SIGGRAPH '96*, pages 217–224, 1996.

[11] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[12] A. Jhala and R. M. Young. A discourse planning approach to cinematic camera control for narratives in virtual environments. In *AAAI'05: Proceedings of the 20th national conference on Artificial intelligence*, pages 307–312. AAAI Press, 2005.

[13] A. Jhala and R. M. Young. Representational requirements for a plan based approach to automated camera control. In *AIIDE*, pages 36–41, 2006.

[14] J. Mascelli. *The Five C's of Cinematography: Motion Picture Filming Techniques*. Cine/Grafic Publications, Hollywood, 1965.

[15] M. Mateas and A. Stern. Facade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference*, 2003.

[16] W. Murch. *In the blink of an eye*. 1986.

[17] R. Ronfard. Automated cinematographic editing tool. Technical report, Xtranormal Technologies, May 7, 2009.

[18] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.

[19] W. Sack and M. Davis. Idic: assembling video sequences from story plans and contentannotations. In *Multimedia Computing and Systems*, pages 30 – 36, 1994.

[20] B. Salt. *Film Style and Technology: History and Analysis (2nd edition)*. Starword, 2003.

[21] A. Salway and M. Graham. Extracting information about emotions in films. In *ACM Conference on Multimedia*, pages 299–302, 2003.

[22] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen. Gaze-based interaction for semi-automatic photo cropping. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 771–780, New York, NY, USA, 2006.

ACM.

[23] H. G. Scott and F. Truffaut. *Hitchcock-Truffaut (Revised Edition)*. Simon and Schuster, 1985.

[24] S. Sharff. *The Elements of Cinema: Toward a Theory of Cinesthetic Impact*. Columbia Press, 1982.

[25] J. Shen, S. Miyazaki, T. Aoki, and H. Yasuda. Intelligent digital filmmaker dmp. In *Computational Intelligence and Multimedia Applications*, pages 272 – 277, 2003.

[26] J. Shen, S. Miyazaki, T. Aoki, and H. Yasuda. Representing digital filmmaking techniques for practical application. In *Information and Knowledge Sharing*, 2003.

[27] M. K. Shirahama and K. Uehara. Video data mining: Extracting cinematic rules from movie. In *Int'l Workshop Multimedia Data Management (MDM-KDD)*, 2003.

[28] T. J. Smith. *An Attentional Theory of Continuity Editing*. PhD thesis, University of Edinburgh, 2005.

[29] S. Sohrabi, J. A. Baier, and S. A. McIlraith. Htn planning with preferences. In *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1790–1797, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[30] H. Zettl. *Sight, Sound, Motion: Applied Media Aesthetics*. Wadsworth Publishing Company, USA, 1999.